PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G07F 7/10, G06K 7/00 — A1

(11) International Publication Number: **WO 99/16030**

(43) International Publication Date: 1 April 1999 (01.04.99)

(21) International Application Number: PCT/IB98/01580

(22) International Filing Date: 18 September 1998 (18.09.98)

(30) Priority Data:

60/059,382	19 September 1997 (19.09.97)	US
09/156,371	17 September 1998 (17.09.98)	US

(81) Designated States: CN, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

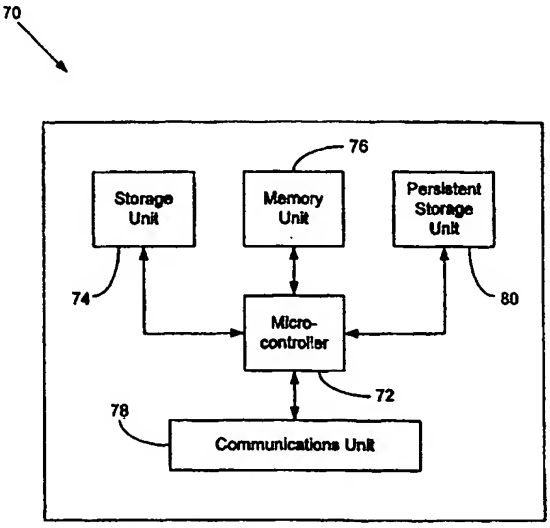
Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(71) Applicant: SCHLUMBERGER INDUSTRIES [FR/FR]; 50, avenue Jean Jaurès, F-92120 Montrouge (FR).

(72) Inventor: MONTGOMERY, Michael, A.; 906 Nelson Ranch Road, Cedar Park, TX 78613 (US).

(74) Agent: LEMOYNE, Didier; Schlumberger Industries, Test & Transactions, Boîte postale 620-04, F-92542 Montrouge Cedex (FR).

(54) Title: SMART CARD APPLICATION-SELECTION



(57) Abstract

A multi-application smart card has a persistent storage unit. The persistent storage unit stores data and/or information that indicates which application is to be run the next time the smart card is inserted into a terminal. The persistent storage unit may indicate the same application, a different user-selected application, or an application-selection application is to be run next. Alternatively, the persistent storage unit may indicate the next application on a predefined list as the application to be run next. An application-selection terminal allows a user to manually select the user-selected application. The terminal may initiate communication with the smart card by sending a special, predefined signal to the smart card.

BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

BEST AVAILABLE COPY

SMART CARD APPLICATION-SELECTION

This application claims the benefit of the filing of U.S. Provisional Patent application No. 60/059,382, filed September 19, 1997.

BACKGROUND OF THE INVENTION

The invention relates to smart cards and, in particular, to smart card application selection in a multi-application smart card.

Smart cards are used for a wide variety of applications including electronic game cards, bank cards, and identification badges. The smart cards are typically encased in a tamper-resistant, plastic or metal housing about the size of a credit card and contain one or more embedded integrated circuit devices. Terminals, such as electronic casinos, automatic teller machines, and ID verification systems, are known which employ one or more smart card interfaces to connect the smart card to the terminal.

Conventional terminals have built-in communications protocols that are used to support the smart card applications. These protocols typically require the smart card, upon receiving a reset from the terminal, to immediately identify itself by providing an Answer To Reset (ATR). Each smart card application may have a different ATR and/or require a different protocol. Therefore, if the terminal does not receive a recognizable ATR, *i.e.*, the terminal does not support the application, it may immediately reject the smart card.

For single-application smart cards, the card simply provides its only ATR, and the terminal either accepts or rejects the card. Referring to FIG. 1, conventional terminals 10, 12, 14, 16 are compatible with and will only recognize certain expected ATRs from smart cards 11, 13, 15, 17, respectively. However, with the advent of multi-application smart cards, even though

the correct card may be inserted, the wrong ATR may be provided to the terminal because each application on the card may have a different ATR and/or communications protocol. The smart card has no opportunity to discover the nature of the terminal before providing the ATR and, therefore, does not know which ATR is required.

One solution is to have a selection mechanism by which the terminal sends the multi-application smart card a "Select" command which selects the particular application to be run. Once selected, information and/or data going to and from the smart card are routed to the selected application until another Select command is received. Cards using this mechanism must have a specific ATR to be recognized by the terminals which support the mechanism.

The selection mechanism works well but has two primary drawbacks. First, most of the presently existing terminals do not support this selection mechanism and will not recognize the associated ATR. Therefore, cards implementing this mechanism will only be compatible with newer terminals that are designed to support application selection and can not be used in the vast base of existing terminals. Second, terminal and terminal device designers and/or software programmers have to expend time, effort and resource to incorporate this selection mechanism into their products. It would be much easier for them to simply use the existing terminal and terminal device specifications.

SUMMARY OF THE INVENTION

The present invention is directed to a method and/or mechanism for selecting the next application to run on a multi-application smart card.

In general, in one aspect, the invention relates to a multi-application smart card that comprises a persistent storage unit that indicates a present application to be run. The smart card runs the application indicated by the persistent storage unit and modifies the persistent storage

unit to indicate a next application to be run. In some embodiments, the smart card may run an application-selection application between each conventional application. The application-selection application allows a user to select which application to run next. The smart card may also select the next application based on a predefined list of applications. The next application may be the same application as the present application. The smart card may use special signaling to activate an application-selection application.

In general, in another aspect, the invention relates to a multi-application smart card comprising means for indicating a present application to be run, means for running the indicated application, and means for modifying the means for indicating to indicate a next application to be run. In one embodiment, the means for modifying modifies the means for indicating to indicate an application-selection application as the next application to be run. In another embodiment, the means for modifying modifies the means for indicating to indicate a next application on a predefined list as the next application to be run. The means for modifying may also modify the means for indicating to indicate the present application as the next application to be run, or a user-selected application as the next application to be run. Upon receiving a predefined signal from a smart card terminal, the means for running may run an application-selection application which allows a user to select which application to run next..

In general, in another aspect, the invention relates to a smart card system comprising a multi-application smart card having a persistent storage unit which designates a next application to be run on the smart card, and an application-selection terminal that allows a user to select which application to be run next on the smart card. In some embodiments, the smart card modifies the persistent storage unit to designate an application-selection application as the

application to be run next. In other embodiments, the terminal modifies the persistent storage unit to designate a user-selected application as the application to be run next.

Advantages of the invention include at least the following: flexibility and ease in selecting one of several applications on a multi-application smart card, and maintaining compatibility with the existing installed base of conventional smart card terminals. This backwards compatibility facilitates use of multi-application smart cards in traditional single application terminals while permitting innovative new applications to be deployed on the same card. Other advantages will become apparent from the following description and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates single-application smart cards and their respective terminals.

FIG. 2 illustrates a multi-application smart card.

FIGS. 3A and 3B are flow charts showing the operation of a smart card wherein the next application to be run is determined according to a predefined list.

FIG. 4 illustrates a multi-application smart card having an application-selection application and an application-selection terminal.

FIGS. 5A-5C are flow charts showing the operation of a smart card wherein an application-selection application is run between each other application.

FIGS. 6A-6D are flow charts showing the operation of a smart card wherein an application-selection is run only upon detection of special signaling.

FIG. 7 illustrates a multi-application smart card of the present invention.

FIG. 8 illustrates an application-selection terminal of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Throughout the description and the figures, elements that are the same will be accorded the same reference numbers.

Referring to FIG. 2, a multi-application smart card 20 contains at least four conventional applications supported by conventional terminals 10, 12, 14, 16, as indicated by the available ATRs. A persistent storage unit 22 on the smart card 20 stores the name, location, or other information indicating which one of the four applications is to be run when the smart card 20 is inserted into the terminals 10, 12, 14, 16. The persistent storage unit 22 may be an EEPROM or FLASH memory, or any other memory which retains its data when the smart card is not powered up. From the above description, it will be appreciated that a mechanism or scheme must be provided to alter or change the value stored in the persistent storage unit 22 to the value of the desired application in order to prevent the wrong application from being run when the smart card 20 is inserted into a terminal.

Referring to FIG. 3A, operation of the multi-application smart card begins with a user inserting the smart card 20 into a conventional or regular terminal (ST30). A transaction is executed between the smart card 20 and the terminal (ST31), and the user removes the smart card 20 from the terminal at the end of the transaction (ST32). Because of the multiple applications on the smart card 20, the user has to determine if the correct application was run (ST33). If it happens that the correct application was run, then the user simply stops or goes about his other business. If the wrong application was run, then the user may reinsert the smart card 20 back into the terminal (ST30).

The key to this embodiment is each time the user executes a transaction (ST31), a different application in a list of applications in the smart card 20 is run. Thus, through a sort of

round-robin process which will be described in more detail below, the correct application may eventually be run by repeated insertion of the smart card 20.

Referring to FIG. 3B, user execution of a transaction (ST31) begins with the smart card 20 receiving a predefined signal, such as a Reset signal, from the terminal (ST31a). Upon receiving the predefined signal, the smart card 20 reads the persistent storage unit 22 to determine the present application to be run (ST31b). The smart card 20 then modifies the persistent storage unit 22 to store the value of the application that is next in line on a predefined list of applications (ST31c). Note that in most embodiments, the predefined list of applications will generally be created and/or modified upon installation or deletion of applications on the smart card, so that the list will track available applications. This application stored in the persistent storage unit 22 will then be the application that is run the next time the smart card 20 is inserted into the terminal. The smart card 20 then runs the present application (ST31d) which sends its ATR to the terminal (ST31e). The terminal then determines if the ATR is one that is recognizable or acceptable to the terminal (ST31f). If it is, then the smart card 20 and the terminal exchange data (ST31g) or otherwise carry out a transaction, and afterwards the transaction is terminated (ST31h). If it is not, then the transaction is terminated right away (ST31h).

This embodiment has the advantage of being able to run any application on the smart card 20 without requiring a special terminal to select the application. However, the process of repeatedly inserting and removing the smart card 20 could be cumbersome if there is a large number of applications on the card. Furthermore, it is possible that some terminals may react adversely to the repeated insertion of a card that is supplying an unrecognized ATR.

In another embodiment, referring to FIG. 4, a smart card 40 has a persistent storage unit 42 and contains a fifth available ATR for a special "application-selection" application which is only supported by a special, application-selection terminal 44. In some embodiments, the application-selection terminal 44 is a Schlumberger PocketDock™ which is a small, portable terminal that can easily be carried around. Clearly, however, any other terminal can be used which supports the application-selection application, including other portable terminals, portable or desktop computers equipped with smart card readers, or any other terminal that supports this application.

When inserted into the terminal 44, the smart card 40 provides the terminal 44 with the ATR for the application-selection application. After the terminal 44 accepts or otherwise acknowledges the ATR, the smart card 40 runs the application-selection application which allows a user to manually select, via the terminal 44, the application that is to be run next. The terminal 44 subsequently modifies the persistent storage unit 42 to indicate the selected application as the application to be run on next insertion.

Referring to FIG. 5A, in one embodiment, operation of the smart card 40 begins by inserting the card into the application-selection terminal 44 (ST50). The user then selects the desired application that is to be run (ST51) and removes the smart card 40 (ST52). Next, the user inserts the smart card into the desired conventional or regular terminal (ST53). The user executes the desired transaction (ST54) and removes the smart card 40 afterwards (ST55).

The key to this embodiment is that the application-selection application is the default application to be run. In order to run any other application, the user must first insert the smart card 40 into the application-selection terminal 44. This allows the user to manually select the

desired application. The user may then insert the smart card 40 into the desired regular terminal confident that the correct application will be run.

Referring to FIG. 5B, user selection of an application (ST51) begins with the smart card 40 receiving a predefined signal from the terminal 44 (ST51a). The smart card 40 then reads the persistent storage unit 42 and finds that the application-selection application is to be run next (ST51b). The smart card 40 then runs the application-selection application (ST51c) which sends its ATR to the terminal 44 (ST51d). The terminal 44 polls, surveys, or otherwise determines the applications that are available on the smart card 40 (ST51e). The terminal 44 then prompts the user to make a selection (ST51f), and the user makes the selection (ST51g). In a preferred embodiment, the terminal 44 then modifies the persistent storage unit 42 to indicate the user-selected application as the application to be run (ST51h). In some embodiments, the terminal 44 may pass the value of the user-selected application to the smart card 40 instead, and the smart card modifies the persistent storage unit 42 accordingly. Afterwards, the transaction is terminated (ST51i).

In this embodiment, referring to FIG. 5C, execution of a transaction (ST54) is in a manner identical to that of the smart card 20 of FIGS. 3A and 3B except the value stored in the persistent storage unit 42 is not determined according to a predefined list. (See FIG. 3B, ST31c). Instead, in this embodiment, the smart card 40 always modifies the persistent storage 42 to indicate the application-selection application as the application to be run next (ST54c).

Note that in this embodiment, in order to use the card in a regular terminal again, even in the very same regular terminal from which the card was just removed, it is necessary to reinsert the smart card 40 back into the terminal 44 and select the desired application again because the application-selection application will have been designated as the next application to run.

Therefore, while the embodiment has the advantage of making it possible for a user to select the desired application without having to upgrade or replace the existing regular terminals, it does require that a terminal 44 be available in order for the user to make the selection.

In another embodiment, as shown in FIG. 6A, the user may insert the smart card 40 into either an application-selection terminal 44 or a regular terminal (ST60) at any time. Once inserted, the smart card 40 relies on special signaling (described below) not normally employed in any conventional terminal to determine if the terminal is an application-selection terminal 44 (ST61). If yes, then the user may select an application to be run (ST62). If no, then the user may execute a regular transaction (ST63). Afterwards, the user removes the smart card 40. When the user is ready to use the smart card again, he may simply insert the smart card again into either types of terminal, and the smart card will run the appropriate application.

Referring to FIG. 6B, the smart card 40 determines whether the terminal is an application-selection terminal (ST61) by, for example, receiving a predefined signal from the terminal (ST61a). The smart card then waits a predefined time period to see if a 2nd predefined signal is received (ST61b). In a preferred embodiment, the smart card 40 uses a microcontroller and/or a counter to measure the elapsed time between two reset signals. If the second reset signal occurs, say, from 0.5 and 1.5 milliseconds after the first, then the smart card determines that the terminal is an application-selection terminal 44. If the second reset occurs in any other time window, the smart card treats the terminal as a regular terminal.

Referring to FIG. 6C, if the terminal is an application-selection terminal 44, the smart card automatically runs the application-selection application (ST62a) regardless of which application is currently designated by the persistent storage unit 42 as the application to run. The application sends its ATR to the terminal 44 (ST62b), and the selection process proceeds as

shown, with the end result being the terminal 44 modifies the persistent storage unit 42 to indicate the user-selected application as the next application (ST62f).

Referring to FIG. 6D, if the terminal is a regular terminal, then the smart card 40 reads the persistent storage unit 42 to determine which application is to be run (ST63a). In a preferred embodiment, the smart card 40 then sets the very same application as the next application to be run (ST63b). This can be done by modifying the persistent storage unit 42 to store the appropriate value, or by simply leaving the current value in the persistent storage unit 42 unchanged. In some embodiments, the smart card 40 may set the next application to be run according to a predefined list. The smart card 40 then runs the present application (ST63c) which sends its ATR to the terminal (ST63d). The smart card 40 and terminal exchange data (ST63e) or otherwise carries out the transaction, and the transaction is terminated (ST63f).

In a typical scenario for this embodiment, the user uses the smart card over and over again in the desired terminal until he is ready to select a different application. He then inserts the smart card into a special application-selection terminal having special signaling capability. The smart card 40, which is equipped with, for example, a microcontroller (not shown) to recognize the special signaling, bypasses the application designated by the persistent storage unit and runs the application-selection application upon receipt and recognition of the special signaling.

This embodiment has the advantage of being able to repeatedly run the same designated application on the smart card without requiring insertion into an application-selection terminal. However, it does require both a specialized smart card and application-selection terminal that can support the special signaling in order to select another application.

A detailed implementation of a preferred embodiment is illustrated by a smart card Java program contained in the Appendix. The program contains a selection application and four

demonstration applications. This program is intended to run in a card which follows the Java card API, such as the Schlumberger CyberFlex™ card, and the application-selection application is designed to be compatible with a Schlumberger PocketDock™.

Referring to FIG. 7, a multi-application smart card 70 comprises a microcontroller 72 which is connected to a storage unit 74 and a memory unit 76. The microcontroller 72 executes smart card software and programs, carries out terminal instructions, and generally manages the flow of data to and from the smart card 70. In some embodiments, the microcontroller 72 may include a microprocessor, a programmable array logic (PAL), an application-specific integrated circuit (ASIC), and/or other integrated circuit devices. The storage unit 74, which may include a read-only memory (ROM), stores the programs and data, including the programs and ATRs for one or more applications, that are needed to operate the smart card 70. The memory unit 76, which may include a random-access-memory (RAM), temporarily stores the programs and data used by the microcontroller 72 during program execution. New or updated programs and data may be downloaded or programmed into the smart card 70 from time to time to upgrade the smart card 70. The smart card 70 also has a communications unit 78 connected thereto which allows the microcontroller 72 to transfer data to and from external devices. In addition, the smart card 70 has a persistent storage unit 80 that stores the name, location, or other information that indicates which application is to be run the next time the smart card 70 is inserted into a terminal. The microcontroller 72, in accordance with the embodiments described above, modifies the persistent storage unit 80 to designate the next application to be run.

Although shown as separate blocks, the microcontroller 72, storage unit 74, memory unit 76, communications unit 78, and persistent memory area 80 may be combined into a single IC device or an otherwise reduced or expanded number of separate IC devices.

Referring now to FIG. 8, an application selection terminal 82 has a smart card interface 84 connected to a microcontroller 86. The interface 84 allows the smart card 80 to be connected to the terminal 82 and facilitates communication between the smart card 70 and the terminal 82. In a preferred embodiment, the interface 84 includes a smart card reader or reader-writer and/or a power supply, such as a battery (not shown), that provides power to the smart card 70. In some embodiments, the interface 84 physically engages the smart card 70. In other embodiments, however, the interface 84 may use inductive, capacitive, or optical coupling, or the interface 84 may use radio frequency signals to connect the smart card 70 to the terminal 82. The microcontroller 86, which may include one or more microprocessor, PAL, ASIC, or other IC devices, executes terminal programs and generally manages the flow of data to and from the smart card 70. A storage unit 88, which may include a hard drive, is connected to the microcontroller 86 and stores the basic programs and data needed to operate the terminal 82. A memory unit 90 temporarily stores the programs and data used by the microcontroller 86 during operation of the terminal 82. In addition, the terminal 82 has a manual selection unit 92, which may be a keypad, keyboard, or other manual selection devices that allow a user to manually enter a selection. The manual selection unit 92 allows the user to select the application that the user desires to run the next time the smart card 70 is inserted into a terminal.

Note that, as is the case with the smart card 70, the microcontroller 86, storage unit 88, and memory unit 90 of the terminal 82 may be combined into a single IC device, or an otherwise reduced or expanded number of IC devices.

Operation of the smart card 70 and terminal 82 is very similar to operation of the embodiments of FIG. 5A-5C and 6A-6D and will not be described here.

It is to be understood that the embodiments described above are merely illustrative and that other arrangements can be devised by one of ordinary skill in the art at the time the invention was made without departing from the scope of the invention. For example, although the special signaling was described as a first and second signal, additional signals and/or other signaling mechanisms such as status lines or communication lines being asserted or toggled, or any other conventional and/or unconventional signaling mechanisms could be employed to trigger the application-selection application. Also, smart card in this invention refers to the microcontroller and/or associated devices, and is not limited to a "card" or any particular form factor. Such a device may be embedded in, for example, a standard credit card format, or may be embedded in a ring, watch, automotive component, or any of a variety of various form factors, without departing from the scope of the invention.

APPENDIX

```
/*
** FlexCash Demo Program - Written by Michael Montgomery 9/17/97
** Copyright (c) 1997 Schlumberger Austin Products Center,
** Schlumberger, Austin, Texas, USA.
**
** Version 1.0
**
** This program was written to test the mechanism of application-selection
** of 4 different demonstration application programs using a Schlumberger
** Pocketdock terminal.
**
** FlexCash State File (0x7701)
**   Offset 0: FlexPoints Purse
**   Offset 1: E-Coins Purse
**   Offset 2: E-Bucks Purse
**   Offset 3: E-Money Purse
**   Offset 4: Current application-selection
*/

public class FlexCash {

  public static void main(String args[]) {
    byte [] Array, Value, Ack;
    byte application, Status;

    Array = new byte[8];

    // Send the first byte of the ATR
    Array[0] = (byte)0x3B;
    _OS.SendMessage(Array,(byte)0x01);

    Ack = new byte[1];

    // Read in the purse values and the application flag
    _OS.SelectFile((short)0x7777);
    _OS.SelectFile((short)0x7701);
    Value = new byte[5];
    _OS.ReadBinaryFile((short)0,(byte)5, Value);
    application = Value[4];

    /* Verify Key 0 */
    Array[0]=(byte)0xAD; Array[1]=(byte)0x9F; Array[2]=(byte)0x61;
    Array[3]=(byte)0xFE; Array[4]=(byte)0xFA; Array[5]=(byte)0x20;
    Array[6]=(byte)0xCE; Array[7]=(byte)0x63;
    _OS.VerifyKey((byte)0, Array, (byte)8);

    /* Set the next application to be the selection application */
    Ack[0] = FlexCashConst.SELECTION;
    Status = _OS.WriteBinaryFile((short)4, (byte)1, Ack);

    switch(application)
    {
```

```
case FlexCashConst.SELECTION:
    // Fill the last part of the FlexCash card ATR and send it
    Array[0] = (byte)0x32; Array[1] = (byte)0x15;
    Array[2] = (byte)0x00; Array[3] = (byte)0x49;
    Array[4] = (byte)0x10;
    _OS.SendMessage(Array,(byte)5);

    // Allocate receive data buffer
    byte[] receiveddatabuffer = new byte[OkiConst.RECEIVED_DATA_LENGTH];

    // Display welcome message
    byte[] dispmapbufferh = new byte[OkiConst.DISP_MAP_CMD_LENGTH];   // Light HELLO
    dispmapbufferh[0] = OkiConst.ISO_ESCAPE;
    dispmapbufferh[1] = OkiConst.INS_LCD_DISPLAY;
    dispmapbufferh[2] = OkiConst.DISPLAY_SEGMENT_MAP;
    dispmapbufferh[3] = OkiConst.PARAM_UNUSED;
    dispmapbufferh[4] = OkiConst.DISP_MAP_DATA_LENGTH;
    dispmapbufferh[5] = (byte)0x00;
    dispmapbufferh[6] = (byte)0x5D;
    dispmapbufferh[7] = (byte)0x03;
    dispmapbufferh[8] = (byte)0x00;
    dispmapbufferh[9] = (byte)0x00;
    dispmapbufferh[10] = (byte)0x07;
    dispmapbufferh[11] = (byte)0x02;
    dispmapbufferh[12] = (byte)0x00;
    dispmapbufferh[13] = (byte)0x00;
    dispmapbufferh[14] = (byte)0x57;
    dispmapbufferh[15] = (byte)0x03;
    dispmapbufferh[16] = (byte)0x00;
    dispmapbufferh[17] = (byte)0x00;
    dispmapbufferh[18] = (byte)0x54;
    dispmapbufferh[19] = (byte)0x01;
    dispmapbufferh[20] = (byte)0x00;
    _OS.SendMessage(dispmapbufferh,OkiConst.DISP_MAP_CMD_LENGTH);
    _OS.GetMessage(receiveddatabuffer,(byte)0x02,OkiConst.ACK_CODE); // Ignore status reply

    // Build key pad command buffers
    byte[] keyscanbuffer = new byte[OkiConst.KEY_SCAN_CMD_LENGTH];
    byte[] keydatabuffer = new byte[OkiConst.KEY_DATA_CMD_LENGTH];

    keyscanbuffer[0] = OkiConst.ISO_ESCAPE;
    keyscanbuffer[1] = OkiConst.INS_KEY_SCAN;
    keyscanbuffer[2] = OkiConst.PARAM_UNUSED;
    keyscanbuffer[3] = OkiConst.PARAM_UNUSED;
    keyscanbuffer[4] = OkiConst.PARAM_UNUSED;

    keydatabuffer[0] = OkiConst.ISO_ESCAPE;
    keydatabuffer[1] = OkiConst.INS_KEY_DATA;
    keydatabuffer[2] = OkiConst.PARAM_UNUSED;
    keydatabuffer[3] = OkiConst.PARAM_UNUSED;
    keydatabuffer[4] = OkiConst.PARAM_UNUSED;

    // Build display command buffers
    byte[] dispfixedbuffer = new byte[OkiConst.DISP_FIXED_CMD_LENGTH]; // Display 1.00
    dispfixedbuffer[0] = OkiConst.ISO_ESCAPE;
```

```
dispfixedbuffer[1] = OkiConst.INS_LCD_DISPLAY;
dispfixedbuffer[2] = OkiConst.DISPLAY_FIXED_POINT;
dispfixedbuffer[3] = OkiConst.PARAM_UNUSED;
dispfixedbuffer[4] = OkiConst.DISP_FIXED_DATA_LENGTH;
dispfixedbuffer[5] = (byte)0x00;
dispfixedbuffer[6] = (byte)0x00;
dispfixedbuffer[7] = (byte)0x00;
dispfixedbuffer[8] = (byte)0x00;
dispfixedbuffer[9] = (byte)0x00;

// Select Next application to run
byte inputkey;
while (true) {
  // Prepare to read keypad
  _OS.SendMessage(keyscanbuffer,OkiConst.KEY_SCAN_CMD_LENGTH);
  _OS.GetMessage(receiveddatabuffer,(byte)0x02,OkiConst.ACK_CODE); // Ignore reply
  // Get control keystroke (ignore all but first)
  _OS.SendMessage(keydatabuffer,OkiConst.KEY_DATA_CMD_LENGTH);
  _OS.GetMessage(receiveddatabuffer,OkiConst.RECEIVED_DATA_LENGTH.
                 OkiConst.ACK_CODE);
  if (receiveddatabuffer[0] != 0)
  {
    dispfixedbuffer[8] = receiveddatabuffer[1];
    _OS.SendMessage(dispfixedbuffer,OkiConst.DISP_FIXED_CMD_LENGTH);
    _OS.GetMessage(receiveddatabuffer,(byte)0x02,OkiConst.ACK_CODE); // Ignore reply
    /* Set the next application to be run */
    Ack[0] = receiveddatabuffer[1];
    Status = _OS.WriteBinaryFile((short)4, (byte)1, Ack);
  }
}


case FlexCashConst.ECOINS:
  // Fill the last part of the FlexCash ECOINS ATR and send it
  Array[0] = (byte)0x32; Array[1] = (byte)0x15;
  Array[2] = (byte)0x00; Array[3] = (byte)0x49;
  Array[4] = FlexCashConst.ECOINS;
  _OS.SendMessage(Array,(byte)5);

  while (true) {
    // Wait for a 5-byte command
    _OS.GetMessage(Array,(byte)0x05,(byte)0x00);
    if(Array[0] == (byte)0xE2) {
      Value[1] = Ack[0] = (byte)(Value[1]-Array[3]);
      Status = _OS.WriteBinaryFile((short)1,(byte)1, Ack);
      _OS.SendStatus(Status);
    } else { // Unknown E-Coins command
      _OS.Execute((short)0,(byte)0);
      _OS.SendStatus((byte)0x92);
      continue;
    }
  }
```

```
case FlexCashConst.EBUCKS:
    // Fill the last part of the FlexCash EBUCKS ATR and send it
    Array[0] = (byte)0x32; Array[1] = (byte)0x15;
    Array[2] = (byte)0x00; Array[3] = (byte)0x49;
    Array[4] = FlexCashConst.EBUCKS;
    _OS.SendMessage(Array,(byte)5);

    while (true) {
      // Wait for a 5-byte command
      _OS.GetMessage(Array,(byte)0x05,(byte)0x00);
      if(Array[0] == (byte)0xE3) {
        Value[3] = Ack[0] = (byte)(Value[2]-Array[3]);
        Status = _OS.WriteBinaryFile((short)2,(byte)1, Ack);
        _OS.SendStatus(Status);
      } else { // Unknown E-Bucks command
        _OS.Execute((short)0,(byte)0);
        _OS.SendStatus((byte)0x93);
        continue;
      }
    }


case FlexCashConst.EMONEY:
    // Fill the last part of the FlexCash EMONEY ATR and send it
    Array[0] = (byte)0x32; Array[1] = (byte)0x15;
    Array[2] = (byte)0x00; Array[3] = (byte)0x49;
    Array[4] = FlexCashConst.EMONEY;
    _OS.SendMessage(Array,(byte)5);

    while (true) {
      // Wait for a 5-byte command
      _OS.GetMessage(Array,(byte)0x05,(byte)0x00);
      if(Array[0] == (byte)0xE4) {
        Value[3] = Ack[0] = (byte)(Value[3]-Array[3]);
        Status = _OS.WriteBinaryFile((short)3,(byte)1, Ack);
        _OS.SendStatus(Status);
      } else { // Unknown E-Money command
        _OS.Execute((short)0,(byte)0);
        _OS.SendStatus((byte)0x94);
        continue;
      }
    }


  case FlexCashConst.LOYALTY:
    // Fill the last part of the FlexCash LOYALTY ATR and send it
    Array[0] = (byte)0x32; Array[1] = (byte)0x15;
    Array[2] = (byte)0x00; Array[3] = (byte)0x49;
    Array[4] = FlexCashConst.LOYALTY;
    _OS.SendMessage(Array,(byte)5);

    while (true) {
      // Wait for a 5-byte command
      _OS.GetMessage(Array,(byte)0x05,(byte)0x00);
```

```
            if(Array[0] == (byte)0xE1) {
              if (Array[1] == (byte)0x02) { // Decrement Loyalty command
                Value[0] = (byte)(Value[0]-Array[3]);
              Status = _OS.WriteBinaryFile((short)0,(byte)1, Value);
              _OS.SendStatus(Status);
              continue;
              }
              if(Array[1] == (byte)0x00) { // Get Values command
              _OS.SendMessage(Ack, (byte)1);
              _OS.SendMessage(Value, (byte)4);
              _OS.SendStatus((byte)0x00);
              continue;
              }
            } else { // Unknown selection command
              _OS.Execute((short)0,(byte)0);
              _OS.SendStatus((byte)0x90);
              continue;
            }
          }


        default:
          _OS.SendStatus((byte)0x95);

      } // end of switch
    } // end of main
} // end of class



public interface FlexCashConst{

  // Constants for Flex Cash program

 static final byte SELECTION  = (byte) 0;
  static final byte ECOINS     = (byte) 1;
  static final byte EBUCKS     = (byte) 2;
  static final byte EMONEY     = (byte) 3;
  static final byte LOYALTY    = (byte) 4;
  static final byte CHALLENGE  = (byte) 7;
  static final byte DEBITFLAG  = (byte) 1;
  static final byte CREDITFLAG = (byte) 2;
}


public interface OkiConst{

 // Constants used for dealing with Pocketdock Reader

   static final byte DISP_FIXED_DATA_LENGTH  = (byte)5;
 static final byte DISP_HEX_DATA_LENGTH    = (byte)4;
 static final byte DISP_MAP_DATA_LENGTH    = (byte)16;
 static final byte DISP_FIXED_CMD_LENGTH  = DISP_FIXED_DATA_LENGTH + (byte)5;
 static final byte DISP_HEX_CMD_LENGTH    = DISP_HEX_DATA_LENGTH + (byte)5;
 static final byte DISP_MAP_CMD_LENGTH    = DISP_MAP_DATA_LENGTH + (byte)5;
 static final byte KEY_SCAN_CMD_LENGTH    = (byte)5;
```

```
static final byte KEY_DATA_CMD_LENGTH    = (byte)5;
static final byte ISO_ESCAPE             = (byte)0xD0;
static final byte INS_LCD_DISPLAY        = (byte)0xE0;
static final byte INS_KEY_SCAN           = (byte)0xE1;
static final byte INS_PIN_ENTRY          = (byte)0xE2;
static final byte INS_KEY_DATA           = (byte)0xE3;
static final byte PARAM_UNUSED           = (byte)0x00;
static final byte DISPLAY_FIXED_POINT    = (byte)0x01;
static final byte DISPLAY_HEXADECIMAL    = (byte)0x02;
static final byte DISPLAY_SEGMENT_MAP    = (byte)0x03;
  static final byte RECEIVED_DATA_LENGTH   = (byte)3;
static final byte ACK_CODE               = (byte)0;
}
```

CLAIMS

What is claimed is:

1. A method of operating a multi-application smart card, comprising:

inserting the smart card into a terminal;

reading a persistent storage unit in the smart card to determine an application to be run;

running the application;

modifying the persistent storage unit to indicate a next application to be run; and

removing the smart card from the terminal.

2. The method of claim 1, wherein modifying includes modifying the persistent storage unit to indicate an application-selection application as the next application to be run.

3. The method of claim 1, wherein modifying includes modifying the persistent storage unit to indicate the present application as the next application to be run.

4. The method of claim 1, wherein modifying includes modifying the persistent storage unit to indicate a user-selected application as the next application to be run.

5. The method of claim 1, wherein modifying includes modifying the persistent storage unit to indicate a next application on a predefined list as the next application to be run.

6. The method of claim 1, further comprising running an application-selection application upon receiving a predefined signal from a terminal.

7. The method of claim 1, further comprising determining applications which are available to be run on the smart card.

8. The method of claim 1, further comprising sending a predefined signal to the smart card.

9. A method of selecting an application to be run in a multi-application smart card, comprising:

inserting a smart card into a terminal;

selecting an application;

configuring the smart card to run the selected application upon next insertion; and

removing the smart card from the terminal.

10. The method of claim 10, wherein the terminal is an application-selection terminal.

11. The method of claim 10, further comprising sending a predefined signal to the smart card.

12. The method of claim 10, further comprising determining which applications are available to be run on the smart card.

13. A smart card capable of running multiple applications, comprising:

a persistent storage unit that indicates a present application to be run; and

a microcontroller that runs the application indicated by the persistent storage unit and modifies the persistent storage unit to indicate a next application to be run.

14. The smart card of claim 13, wherein the microcontroller modifies the persistent storage unit to indicate an application-selection application as the next application to be run.

15. The smart card of claim 13, wherein the microcontroller modifies the persistent storage unit to indicate a next application on a predefined list as the next application to be run.

16. The smart card of claim 13, wherein the microcontroller modifies the persistent storage unit to indicate the present application as the next application to be run.

17. The smart card of claim 13, wherein the microcontroller modifies the persistent storage unit to indicate a user-selected application as the next application to be run.

18. The smart card of claim 13, wherein upon receiving one or more predefined signals from a smart card terminal, the microcontroller runs an application-selection application.

19. The smart card of claim 18, wherein the predefined signals includes a first reset followed by a second reset after a predefined interval.

20. A smart card capable of running multiple applications, comprising:

means for indicating a present application to be run;

means for running the indicated application; and

means for modifying the means for indicating to indicate a next application to be run.

21. The smart card of claim 20, wherein the means for modifying modifies the means for indicating to indicate an application-selection application as the next application to be run.

22. The smart card of claim 20, wherein the means for modifying modifies the means for indicating to indicate a next application on a predefined list as the next application to be run.

23. The smart card of claim 20, wherein the means for modifying modifies the means for indicating to indicate the present application as the next application to be run.

24. The smart card of claim 20, wherein means for modifying modifies the means for indicating to indicate a user-selected application as the next application to be run.

25. The smart card of claim 20, wherein upon receiving a predefined signal from a smart card terminal, the means for running runs an application-selection application.

26. The smart card of claim 25, wherein the predefined signal includes a first reset followed, after a predefined interval, by a second reset.

27. A smart card application selection terminal, comprising:

a smart card interface that connects a multi-application smart card to the terminal; and

a manual selection unit for manually selecting an available application as a next application to be run on the multi-application smart card.

28. The terminal of claim 27, further comprising a microcontroller that determines which applications are available to be run on the multi-application smart card.

29. The terminal of claim 27, wherein the microcontroller initiates communication with the multi-application smart card by sending a predefined signal to the smart card.

30. The terminal of claim 27, wherein the microcontroller communicates, to the smart card, the application selected as the next application to be run.

31. A smart card application-selection terminal, comprising:

means for interfacing with a multi-application smart card; and

means for manually selecting an available application as a next application to be run on the multi-application smart card.

32. The terminal of claim 31, further comprising means for initiating communication with the smart card by sending a predefined signal to the smart card.

33. The terminal of claim 31, further comprising means for communicating to the smart card which application was selected as the next application to be run.

34. The terminal of claim 31, further comprising means for determining applications which are available to be run on the multi-application smart card.

35. A smart card system, comprising:

a multi-application smart card having a persistent storage unit that designates a next application to be run on the smart card; and

an application-selection terminal that allows a user to select which application to be run next on the smart card.

36. The system of claim 35, wherein the smart card modifies the persistent storage unit to designate an application-selection application as the application to be run next.

37. The system of claim 35, wherein the terminal modifies the persistent storage unit to designate a user-selected application as the application to be run next.

38. The smart card of claim 13, wherein the smart card may assume any form factor.

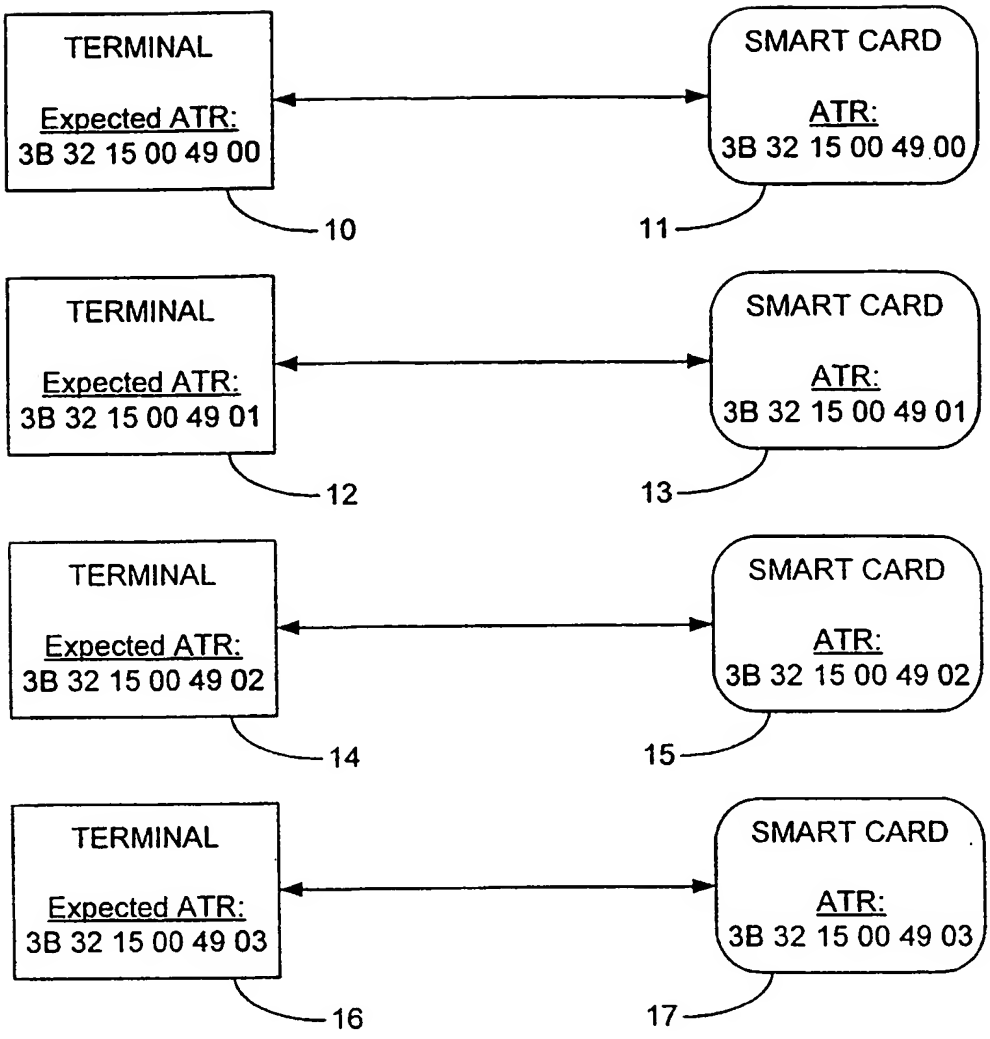


FIG. 1
(PRIOR ART)

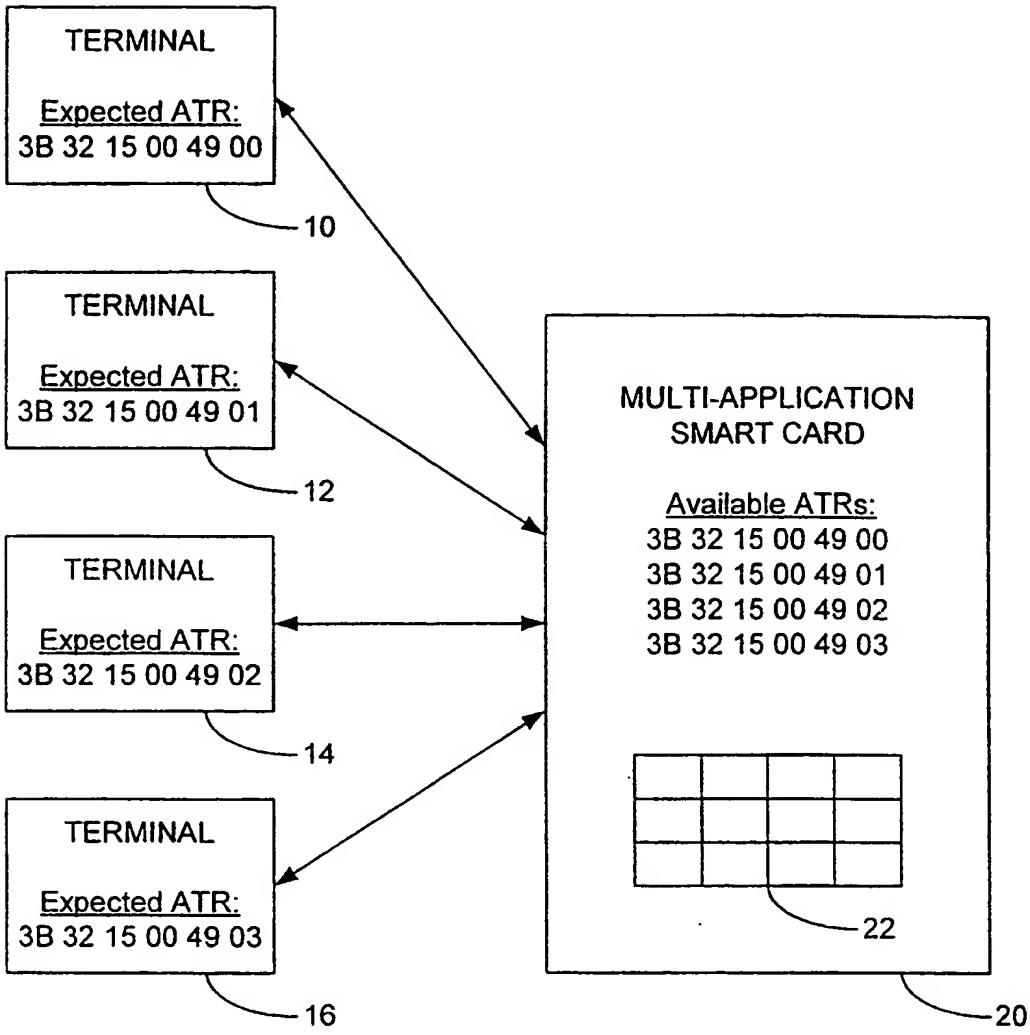

TERMINAL
Expected ATR:
3B 32 15 00 49 00
10
TERMINAL
Expected ATR:
3B 32 15 00 49 01
12
TERMINAL
Expected ATR:
3B 32 15 00 49 02
14
TERMINAL
Expected ATR:
3B 32 15 00 49 03
16
MULTI-APPLICATION
SMART CARD
Available ATRs:
3B 32 15 00 49 00
3B 32 15 00 49 01
3B 32 15 00 49 02
3B 32 15 00 49 03
22
20

FIG. 2

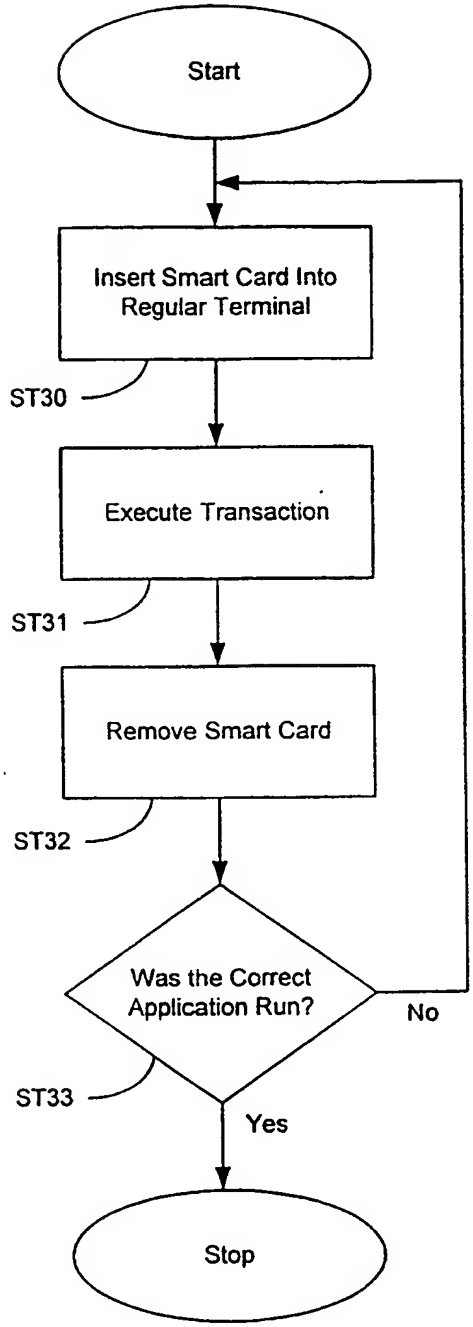

Start
Insert Smart Card Into Regular Terminal
ST30
Execute Transaction
ST31
Remove Smart Card
ST32
Was the Correct Application Run?
No
ST33
Yes
Stop

FIG. 3A

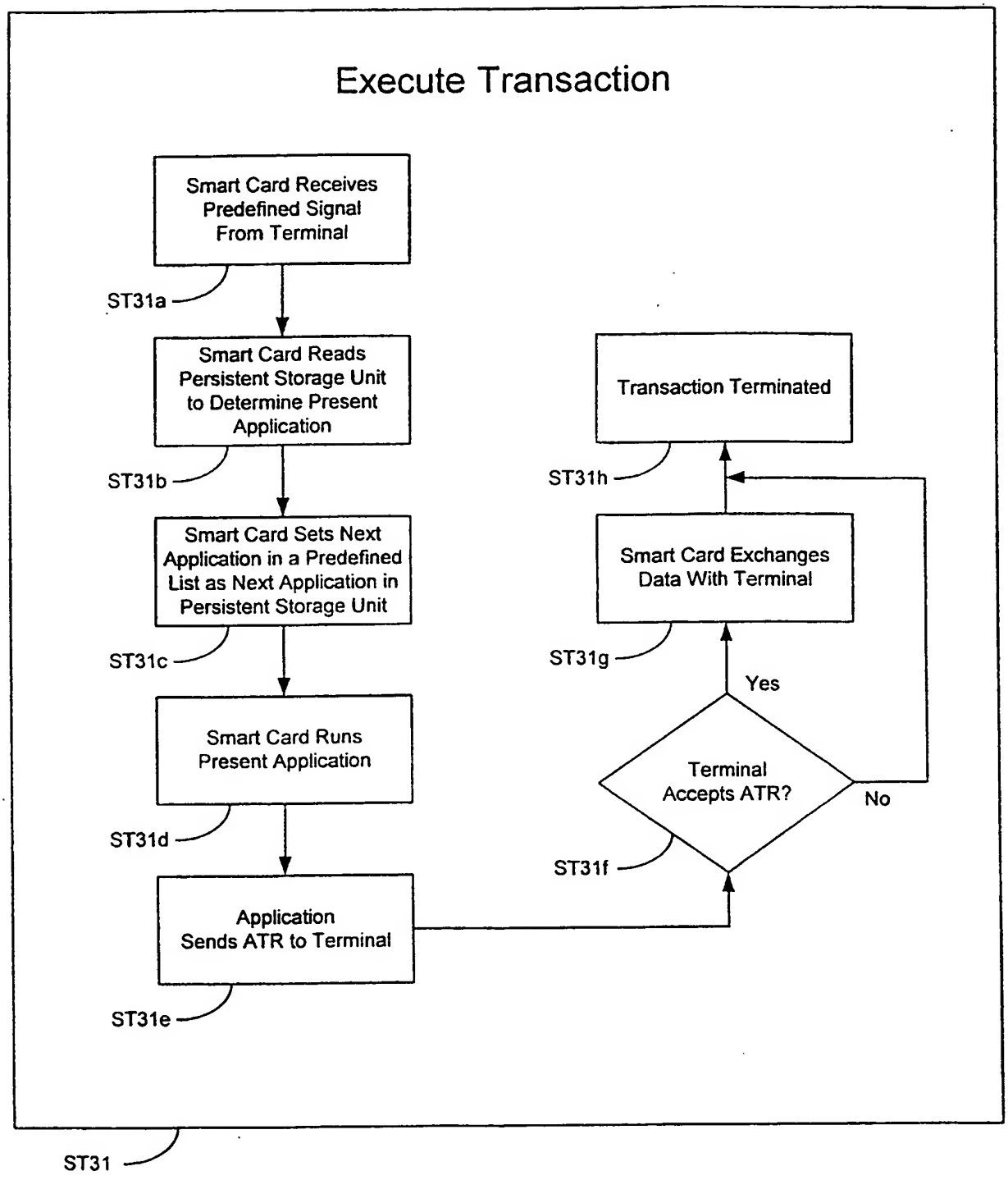

Execute Transaction
Smart Card Receives Predefined Signal From Terminal
ST31a
Smart Card Reads Persistent Storage Unit to Determine Present Application
ST31b
Smart Card Sets Next Application in a Predefined List as Next Application in Persistent Storage Unit
ST31c
Smart Card Runs Present Application
ST31d
Application Sends ATR to Terminal
ST31e
Terminal Accepts ATR?
ST31f
No
Yes
Smart Card Exchanges Data With Terminal
ST31g
Transaction Terminated
ST31h
ST31

FIG. 3B

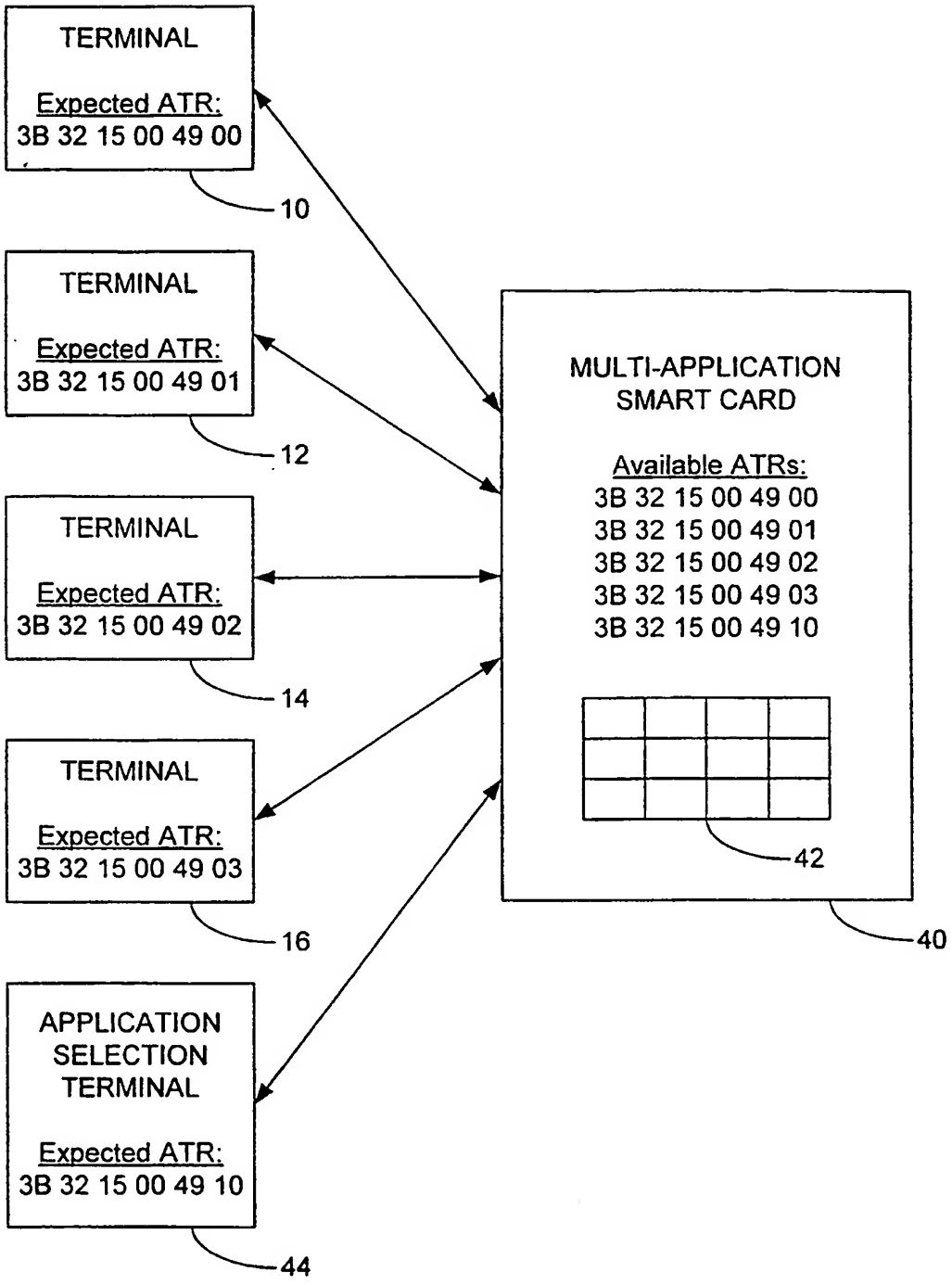

TERMINAL
Expected ATR:
3B 32 15 00 49 00
10
TERMINAL
Expected ATR:
3B 32 15 00 49 01
12
TERMINAL
Expected ATR:
3B 32 15 00 49 02
14
TERMINAL
Expected ATR:
3B 32 15 00 49 03
16
APPLICATION SELECTION TERMINAL
Expected ATR:
3B 32 15 00 49 10
44
MULTI-APPLICATION SMART CARD
Available ATRs:
3B 32 15 00 49 00
3B 32 15 00 49 01
3B 32 15 00 49 02
3B 32 15 00 49 03
3B 32 15 00 49 10
42
40

FIG. 4

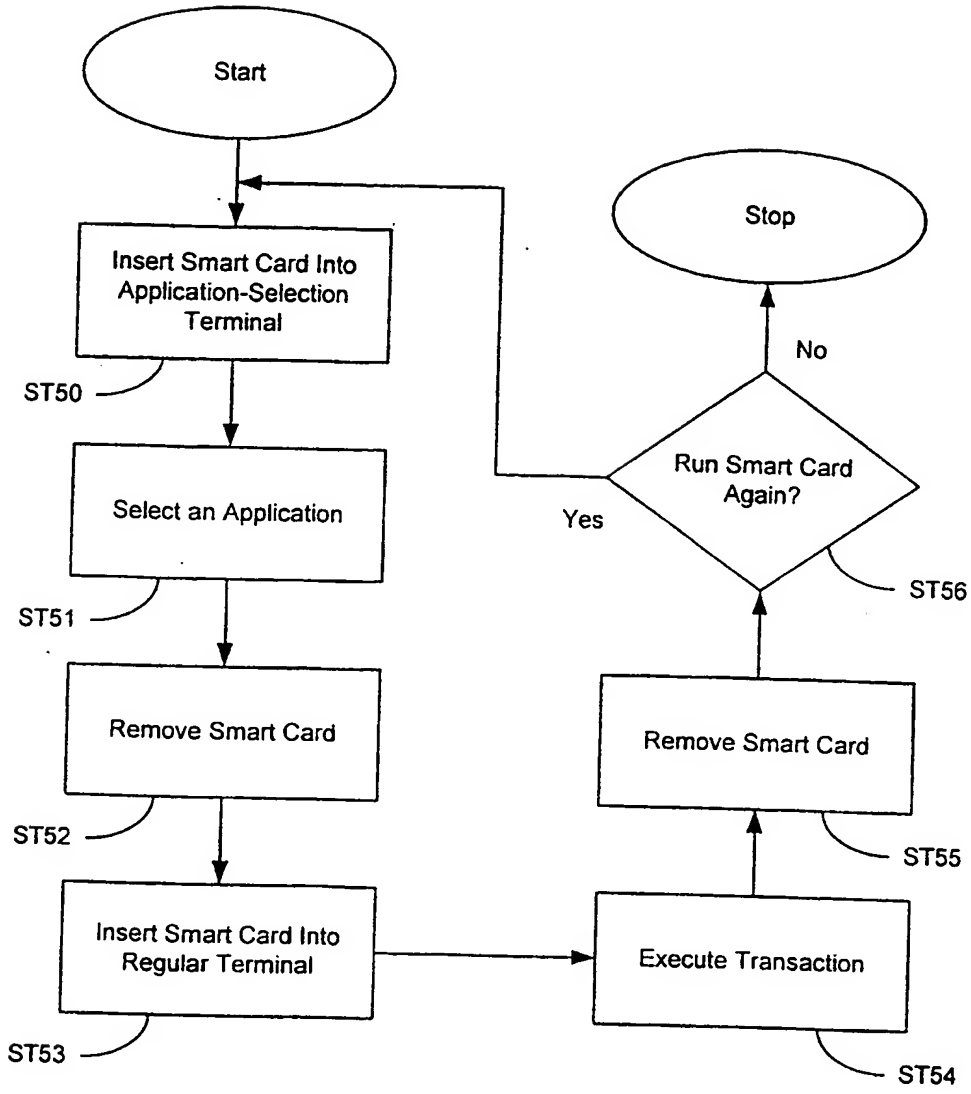

Start
Insert Smart Card Into Application-Selection Terminal
ST50
Select an Application
ST51
Remove Smart Card
ST52
Insert Smart Card Into Regular Terminal
ST53
Execute Transaction
ST54
Remove Smart Card
ST55
Run Smart Card Again?
ST56
Yes
No
Stop

FIG. 5A

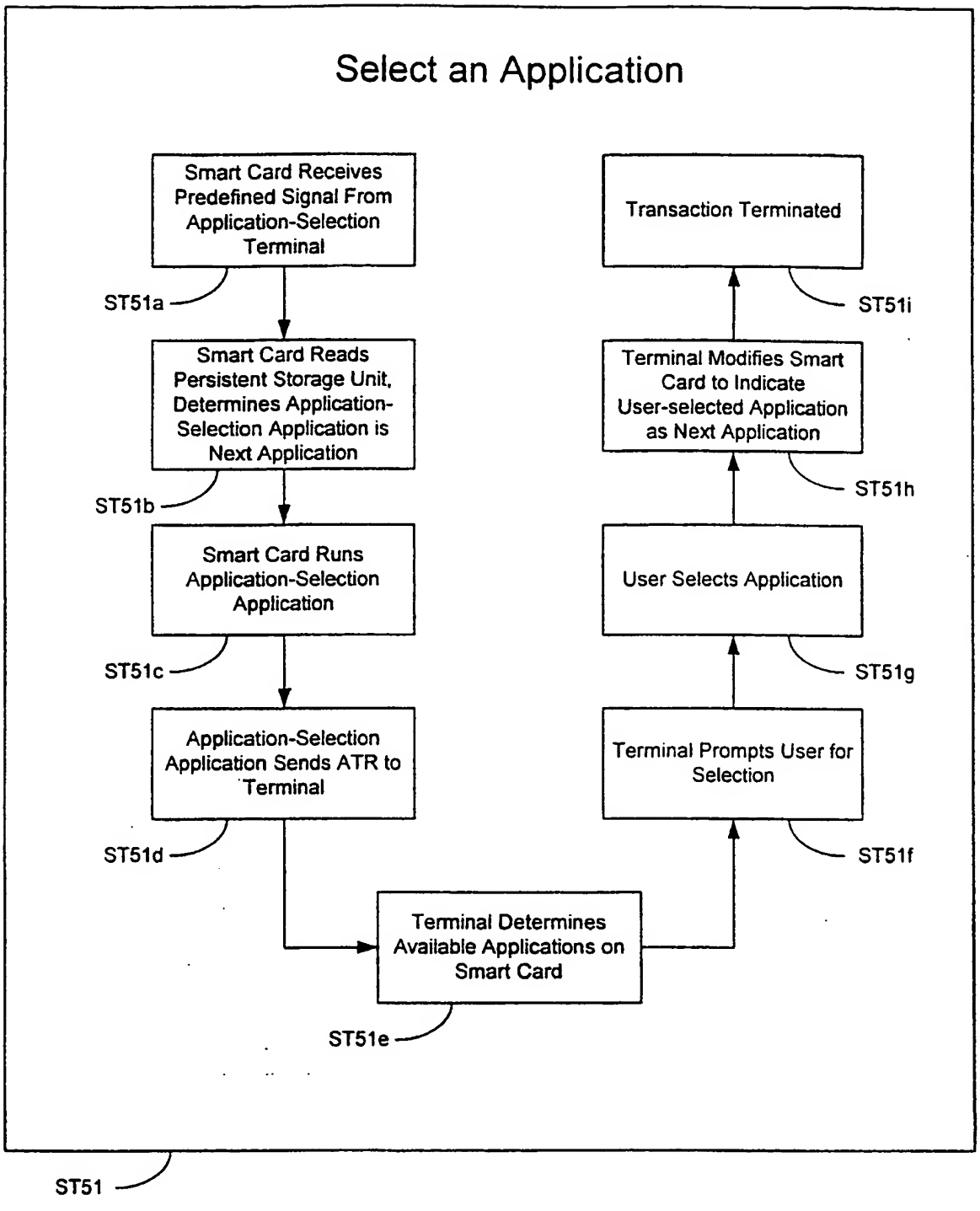

Select an Application
Smart Card Receives Predefined Signal From Application-Selection Terminal
ST51a
Smart Card Reads Persistent Storage Unit, Determines Application-Selection Application is Next Application
ST51b
Smart Card Runs Application-Selection Application
ST51c
Application-Selection Application Sends ATR to Terminal
ST51d
Terminal Determines Available Applications on Smart Card
ST51e
Terminal Prompts User for Selection
ST51f
User Selects Application
ST51g
Terminal Modifies Smart Card to Indicate User-selected Application as Next Application
ST51h
Transaction Terminated
ST51i
ST51

FIG. 5B

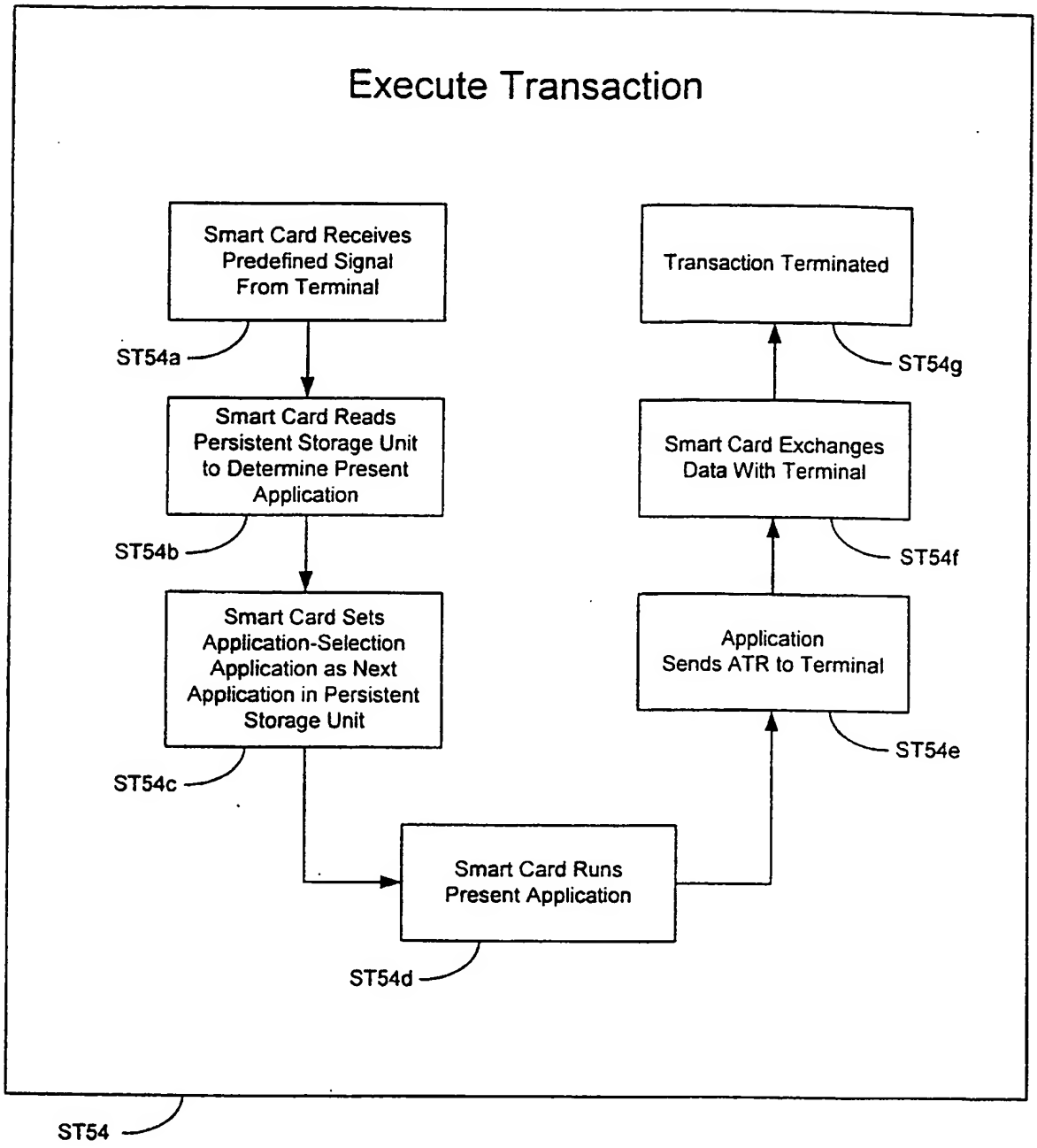

Execute Transaction
Smart Card Receives Predefined Signal From Terminal
ST54a
Smart Card Reads Persistent Storage Unit to Determine Present Application
ST54b
Smart Card Sets Application-Selection Application as Next Application in Persistent Storage Unit
ST54c
Smart Card Runs Present Application
ST54d
Application Sends ATR to Terminal
ST54e
Smart Card Exchanges Data With Terminal
ST54f
Transaction Terminated
ST54g
ST54

FIG. 5C

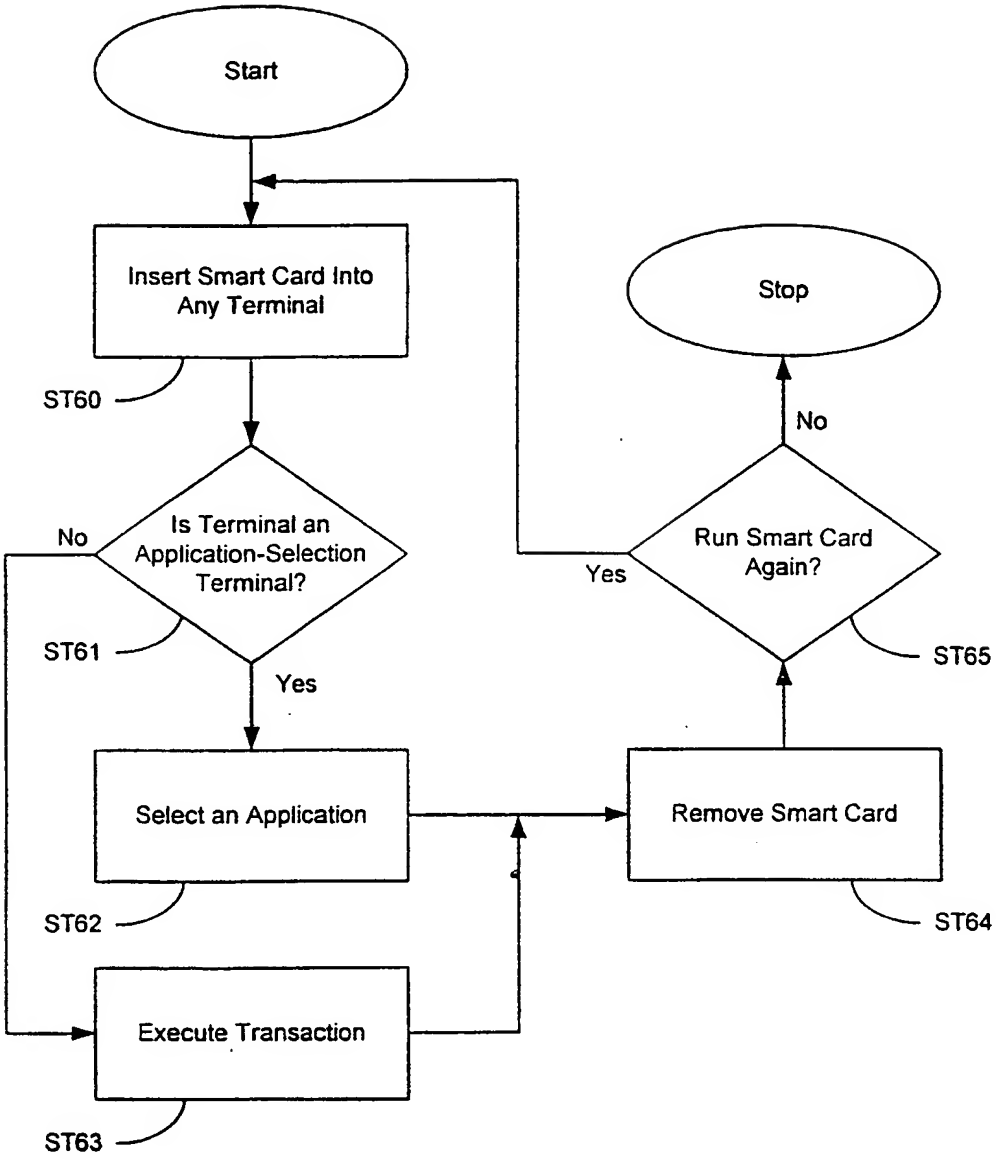

Start
Insert Smart Card Into Any Terminal
ST60
Is Terminal an Application-Selection Terminal?
No
ST61
Yes
Select an Application
ST62
Execute Transaction
ST63
Remove Smart Card
ST64
Run Smart Card Again?
Yes
ST65
No
Stop

FIG. 6A

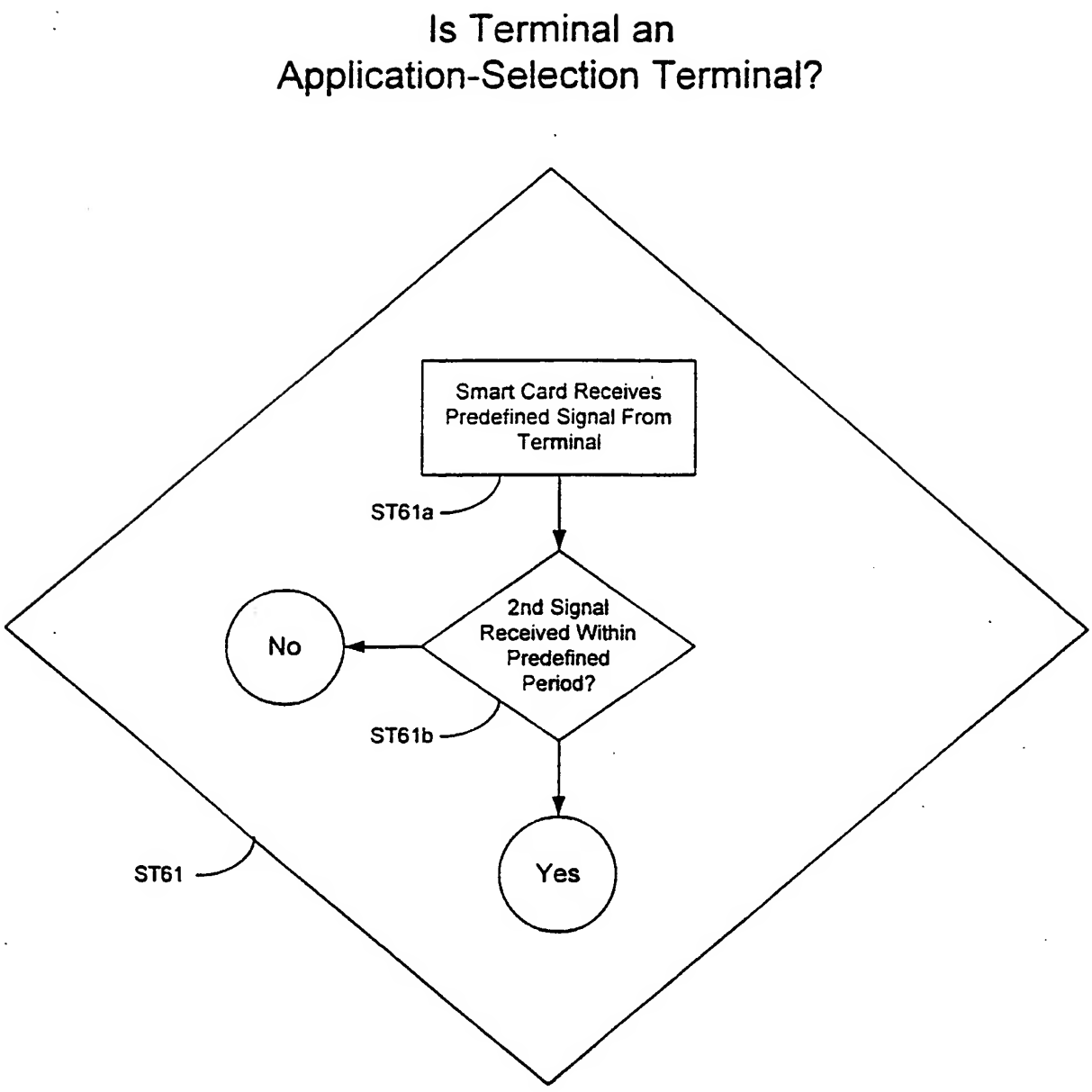

Is Terminal an Application-Selection Terminal?
Smart Card Receives Predefined Signal From Terminal
ST61a
2nd Signal Received Within Predefined Period?
No
ST61b
Yes
ST61

FIG. 6B

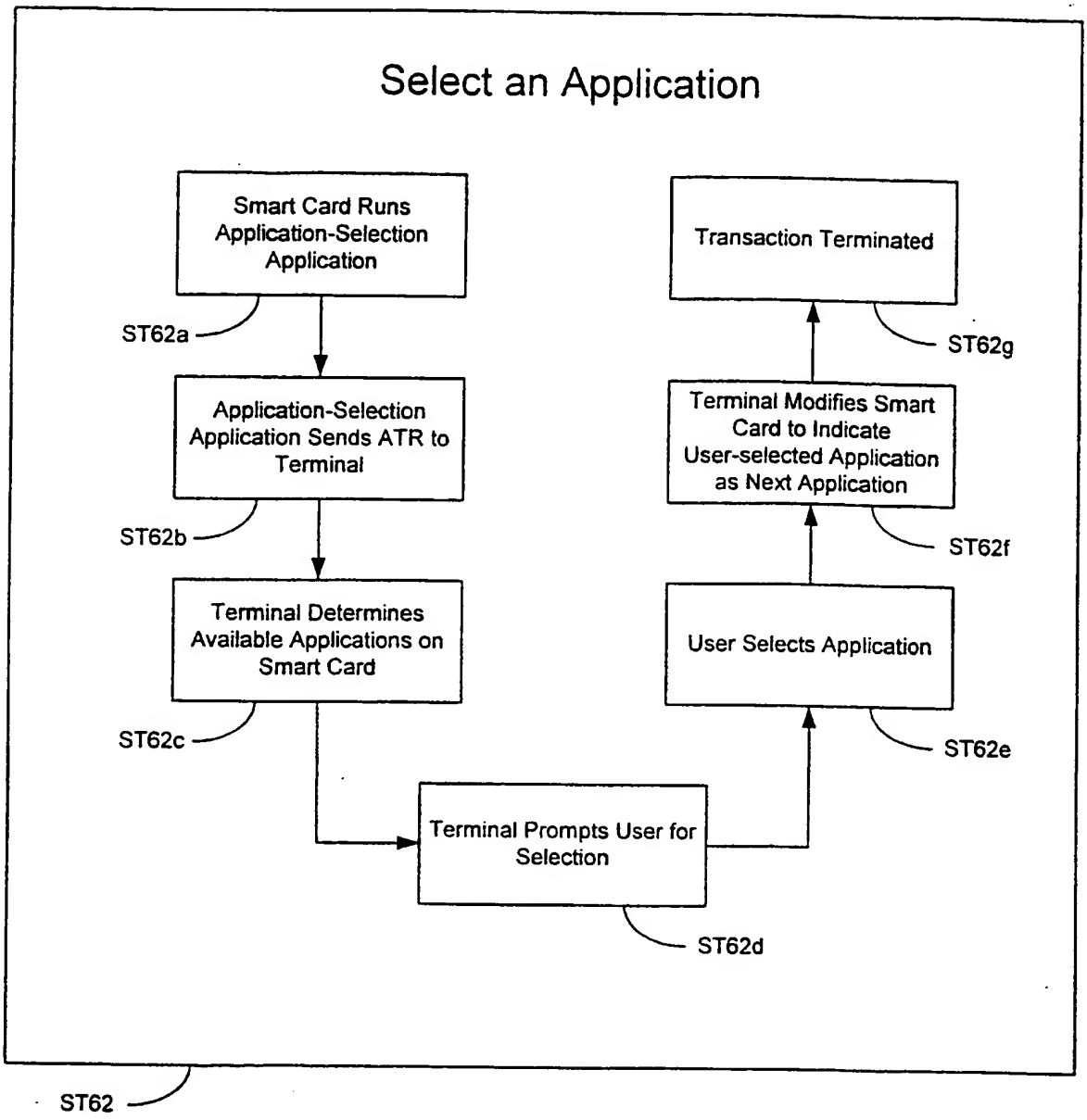

Select an Application
Smart Card Runs Application-Selection Application
ST62a
Application-Selection Application Sends ATR to Terminal
ST62b
Terminal Determines Available Applications on Smart Card
ST62c
Terminal Prompts User for Selection
ST62d
User Selects Application
ST62e
Terminal Modifies Smart Card to Indicate User-selected Application as Next Application
ST62f
Transaction Terminated
ST62g
ST62

FIG. 6C

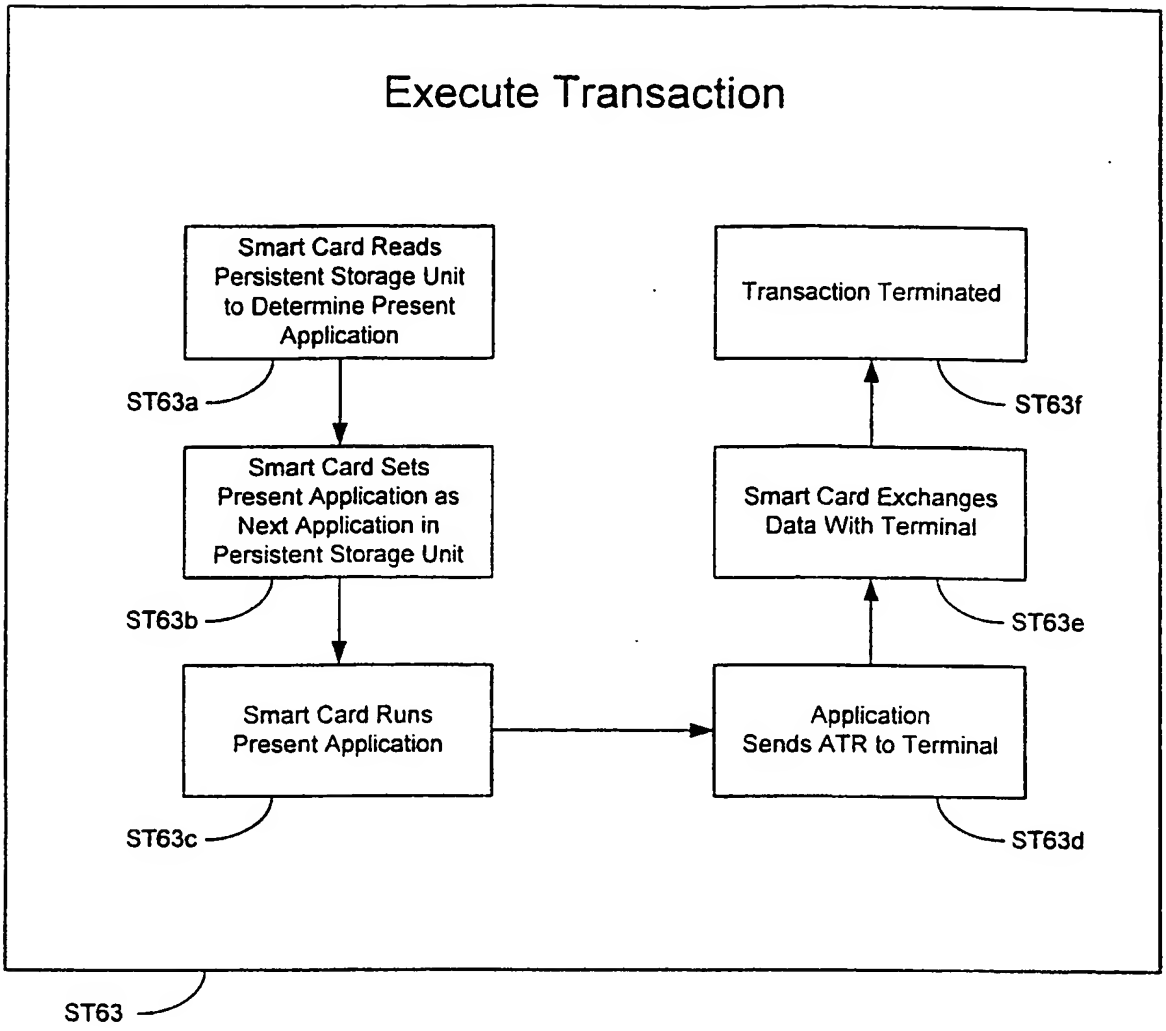

Execute Transaction
Smart Card Reads Persistent Storage Unit to Determine Present Application
ST63a
Smart Card Sets Present Application as Next Application in Persistent Storage Unit
ST63b
Smart Card Runs Present Application
ST63c
Application Sends ATR to Terminal
ST63d
Smart Card Exchanges Data With Terminal
ST63e
Transaction Terminated
ST63f
ST63

FIG. 6D

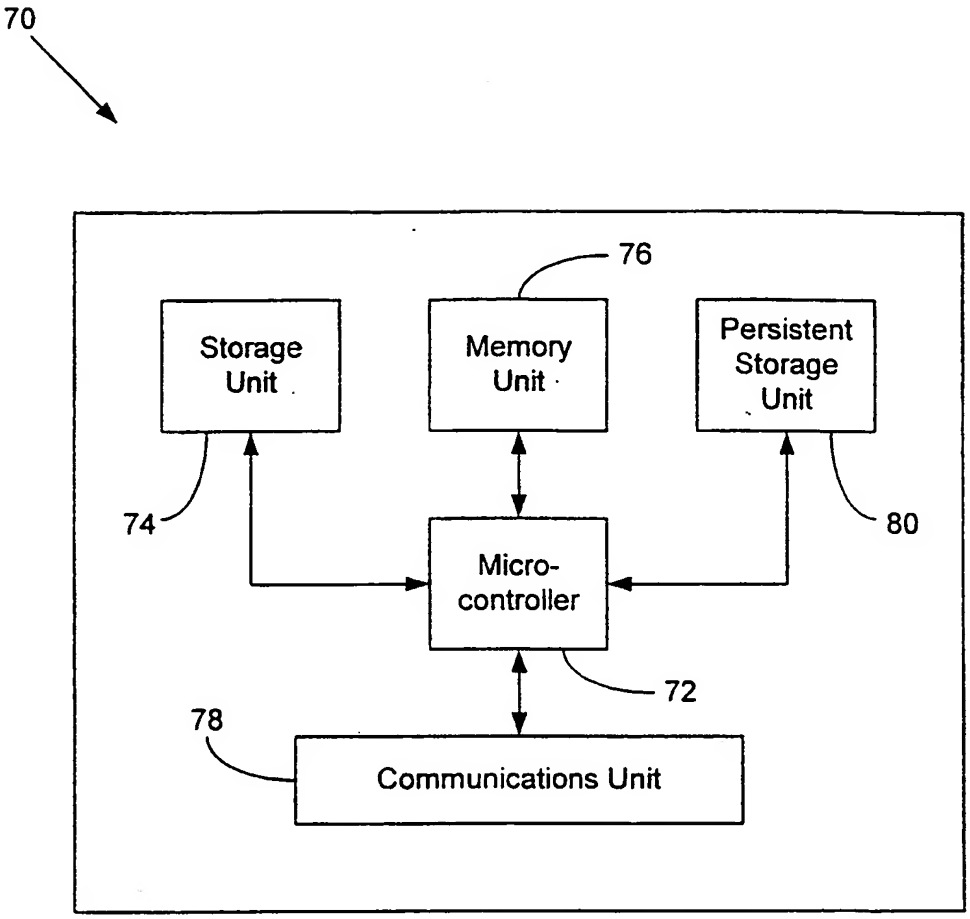

70
76
Storage Unit
Memory Unit
Persistent Storage Unit
74
80
Micro-controller
72
78
Communications Unit

FIG. 7

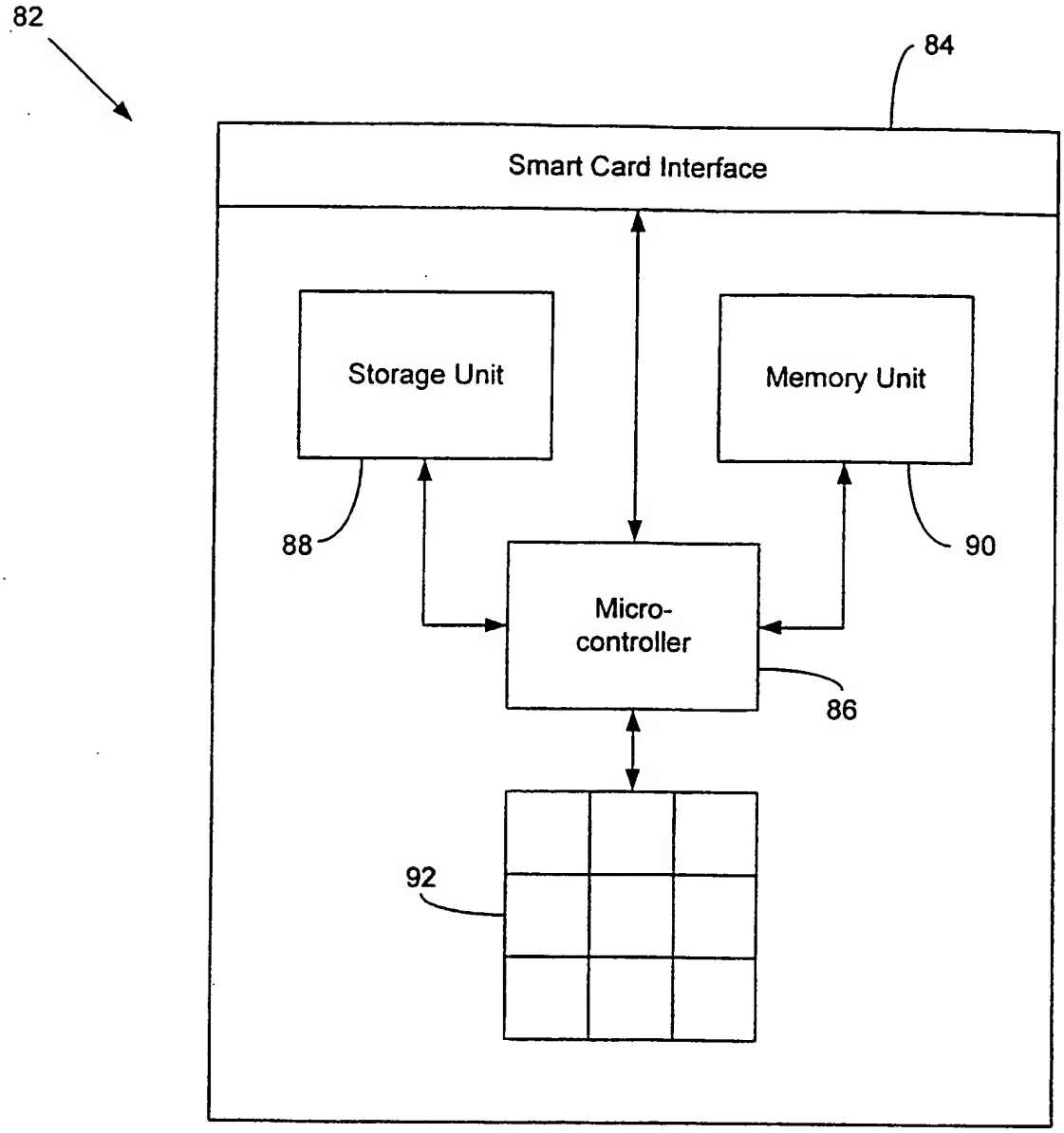

82
84
Smart Card Interface
Storage Unit
Memory Unit
88
90
Micro-
controller
86
92

FIG. 8

INTERNATIONAL SEARCH REPORT

rnational Application No
PCT/IB 98/01580

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G07F7/10 G06K7/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G07F G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 717 381 A (MASTERCARD INTERNATIONAL) 19 June 1996 see abstract; claims; figures 1-5	1,2,4,7, 9,10, 12-14, 17,20, 21,24, 27-38
A	US 5 581 708 A (Y. IIJIMA) 3 December 1996 see abstract; claims; figures see column 5, line 28 - column 8, line 41	1,5-9, 11-13, 15,20, 22,25, 27-29, 31,32, 34,35

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 February 1999	19/02/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	David, J

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IB 98/01580

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 159 651 A (OMRON TATEISI ELECTRONICS) 30 October 1985	
A	EP 0 513 507 A (TOSHIBA) 19 November 1992	

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

national Application No
PCT/IB 98/01580

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0717381 A	19-06-1996	US 5705798 A	06-01-1998
		AU 692584 B	11-06-1998
		AU 3205695 A	03-07-1996
		CA 2207970 A	20-06-1996
		WO 9618979 A	20-06-1996
US 5581708 A	03-12-1996	JP 6276249 A	30-09-1994
		FR 2703167 A	30-09-1994
EP 0159651 A	30-10-1985	JP 60220464 A	05-11-1985
		JP 1796069 C	28-10-1993
		JP 5003619 B	18-01-1993
		JP 60221870 A	06-11-1985
		JP 1796070 C	28-10-1993
		JP 5003620 B	18-01-1993
		JP 60221871 A	06-11-1985
		JP 1796071 C	28-10-1993
		JP 5003621 B	18-01-1993
		JP 60229169 A	14-11-1985
		JP 1796074 C	28-10-1993
		JP 5003622 B	18-01-1993
		JP 60233771 A	20-11-1985
		JP 1796075 C	28-10-1993
		JP 5003623 B	18-01-1993
		JP 60237575 A	26-11-1985
		DE 3586367 T	27-08-1992
		US 4709137 A	24-11-1987
EP 0513507 A	19-11-1992	JP 4321192 A	11-11-1992
		DE 69216907 D	06-03-1997
		DE 69216907 T	04-09-1997
		US 5349649 A	20-09-1994

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.